

S/N 09/520,405

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Michael G. Martinek et al.

Examiner: S. Ashburn

Serial No.: 09/520,405

Group Art Unit: 3713

Filed: March 8, 2000

Docket: PA0390.ap.US

Title: COMPUTERIZED GAMING SYSTEM METHOD AND APPARATUS

**APPEAL BRIEF TO THE BOARD OF
PATENT APPEALS AND INTERFERENCES OF THE
UNITED STATES PATENT AND TRADEMARK OFFICE**

MAIL STOP: APPEAL BRIEF - PATENTS

P.O. BOX 1450

Commissioner for Patents

Alexandria, VA22313-1450

RECEIVED

FEB 17 2004

TECHNOLOGY CENTER R3700

Sir:

This is an appeal from the Office Action mailed on September 8, 2003 rejecting claims 1-38 and 47-57, all of the claims in the Application for the fifth time, without making the rejection final. All other claims are withdrawn/cancelled as drawn to a non-elected invention.

This Brief is being filed in triplicate along with authorization to debit \$320.00 to Deposit Account No. 50-1391 to cover the fee for the appeal. Appellants request the opportunity for a personal appearance before the Board of Appeals to argue the issues of this appeal. The fee for the personal appearance will be timely paid upon receipt of the Examiner's Answer.

CERTIFICATE UNDER 37 C.F.R. 1.8: The undersigned hereby certifies that this Transmittal Letter and the paper, as described herein, are being deposited in the United States Postal Service, as first class mail, with sufficient postage, in an envelope addressed to: MAIL STOP: APPEAL BRIEFS - PATENT, P.O. BOX 1450, Commissioner for Patents, Alexandria, VA 22313-1450 9 February 2004.

Mark A. Litman
Name

Signature

02/13/2004 DTESSEM1 00000014 501391 09520405.

01 FC:2402 165.00 DA

TABLE OF CONTENTS

Page No.

Real Party in Interest.....	3
Related Appeals and Interferences	4
Status of Claims.....	5
Status of Amendments.....	6
Summary of the Invention.....	7
Issues on Appeal.....	8-10
Grouping of Claims.....	11-12
Arguments.....	13-32
Conclusion.....	33
Appendix.....	34-45

REAL PARTY IN INTEREST

The real party in interest is International Game Technology, Inc. a corporation of the State of Nevada having a place of business at 9295 Prototype Dr. Reno, NV 89521. This application was originally filed as an application assigned to Shuffle Master, Inc., but the application has been sold and assigned in its entirety to International Gaming Technologies, Inc.

RELATED APPEALS AND INTERFERENCES

Appellants do not know of any other pending U.S. Patent Applications that are on appeal which have issues that overlap with the issues in this Appeal. No Interference proceedings before the U.S. Patent and Trademark Office are known by Appellants to have any substantive relationship to the subject matter of this Appeal.

STATUS OF CLAIMS

As no final rejection has been made on these claims, there is no amendment after final rejection. Claims 1-38 and 47-57, all of the claims in the Application, have been rejected under 35 USC 103(a) over a combination of references. As four separate Office Actions rejecting the claims have been made, an appeal was deemed appropriate and is authorized under the statutes.

STATUS OF AMENDMENTS

All amendments made to date have been entered without objection. As there has been no final rejection, there has been no amendment submitted under 37 CFR 1.116, but only under 37 CFR 1.111.

SUMMARY OF THE INVENTION

The present invention in various embodiments provides a computerized wagering game method and apparatus that features a customized operating system kernel with selected device handlers that are disabled or removed, features a system handler application that loads and executes a single gaming program object at any time, and features nonvolatile storage that facilitates sharing of information between gaming program objects. The system handler of some embodiments further provides an API library of functions callable from the gaming program objects, and facilitates the use of callback functions on change of data stored in nonvolatile storage. The nonvolatile storage also provides a nonvolatile record of the state of the computerized wagering game, providing protection against loss of the game state due to power loss. The system handler application in various embodiments includes an event handler, providing an interface to selected hardware and the ability to monitor hardware-related events. (Page 7, lines 2-15). A specific aspect of the improved method and apparatus is the following language or its functional equivalent recited in all claims on Appeal:

that a “program object calls up an Application Program Interface.”

In contrast, all prior art systems shown in references used in the rejection have passive shared objects that are called through an API. Our claims require an active program object that functions to call a system handler application. (Page 10, line 19 – page 11, line 22).

ISSUES ON APPEAL

Remarks Concerning the Rejections

In the Office Action mailed on March 20, 2003, the U.S. PTO:

a) Asserted that the “proposed drawing correction and/or the proposed substitute sheets of drawings, filed December 9, 2002 have been accepted.” The statement then continues that “A proper drawing correction or corrected drawings are required in reply to the Office Action to avoid abandonment of the application.” As the formal drawings with labels have already been submitted and accepted and there are no more outstanding objections to the drawings made, it is assumed that all formality requirements with the drawings have been addressed. This amendment is being filed well in advance of the statutory date so that sufficient time to correct any issues may still be available. **As these comments were not addressed or repeated in the office action, it is assumed that the drawings are in proper form and have been accepted.**

Issues on Appeal – The Generic Issue on Appeal is Whether the Following Rejections May Be sustained in View of the Substantive Arguments Provided herein against the Underlying Bases of the Rejections

b) Claims 1-3, 11-13, 16, 19, 29, 31, 34-37, 39-43, 48 and 50-54 are rejected under 35 USC 103(a) as unpatentable over Bunnell (U.S. Patent No. 6,075,939).

c) Claims 13, 14, 18-27 and 49 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Fullerton (U.S. Patent No. 4,607,844).

d) Claims 15 and 49 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Fullerton (U.S. Patent No. 4,607,844) (as above) when further considered with Pascal et al. (U.S. Patent No. 5,791,851).

e) Claims 28-33 have been rejected under 35 U.S.C. 103(a) as unpatentable over Houriet, Jr. et al. (U.S. Patent No. 5,575,717) in view of Mitchell et al. (U.S. Patent No. 5,872,973).

f) Claims 7, 8, and 14 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of David A. Rusling, *The Linux Kernel*, (Hereinafter "Rusling").

g) Claims 9, 10, 17, 38, 44 and 47 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Pascal (as applied to claims 22, 23) in further view of Bock (U.S. Patent No. 5,155,856) and Davis (U.S. Patent No. 6,401,208).

h) Claims 45, 46 and 50-51 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Wiltshire (U.S. Patent No. 6,409,602).

i) Claims 4, 5, 38 and 55-57 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Arbaugh et al. (U.S. Patent No. 6,185,678).

j) Claim 6 has been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Arbaugh et al. (U.S. Patent No. 6,185,678) (as above) in further view of Pascal.

In addition to these generic issues, the specific issue presented is that where a) the rejection itself admits that the primary reference fails to establish utility of the teachings of that reference in the field of gaming, b) multiple elements and limitations of the invention are acknowledged by the Office Action to be missing from the Bunnell et al. reference, and c) the primary reference (Bunnell) fails to describe, suggest or enable a system handler to dynamically link to at least one gaming program object, and there is no motivation for obviousness from the four secondary references used in the rejection, can obviousness be established. The four secondary references fail to teach these limitations and in some cases even fail to teach the specific limitations for which they were cited in the Office Action.

GROUPING OF CLAIMS

The following grouping of claims is made in compliance with the requirements of 37 C.F.R. 1.191 for the content of an Appeal Brief. The following grouping of claims is made to expedite this Appeal and narrow issues, and is not intended to waive or limit the right of the Applicants to enforce and defend claims separately, even though they are grouped for convenience in this Appeal.

With regard to each of the following rejections, the grouping of claims is provided:

b) Claims 1-3, 11-13, 16, 19, 29, 31, 34-37, 39-43 (cancelled), 48 and 50-54 are rejected under 35 USC 103(a) as unpatentable over Bunnell (U.S. Patent No. 6,075,939).

Claims 1, 2, 16, 29, 31, 35, 48, 50, 51, and 53-54 shall stand or fall with the patentability of claim 1.

Claim 3 shall stand or fall with itself for patentability, that claim specifically reciting loading and unloading and executing specific functions within a gaming environment.

Claims 11-12 shall stand or fall with the patentability of claim 11, which recites calling specific gaming programs from an API.

Claim 13 shall stand or fall by itself, that claim reciting a specific procedure in the performance of the process by first calling functions from the API, loading a first program, executing the first program, unloading the first program, then loading a second program.

Claim 19 shall stand or fall by itself. In addition to all of the limitations of claim 13, this claim additionally requires that the second (and therefore different function) also be called from within the API.

Claims 34 and 36-37 shall stand or fall with the patentability of claim 34, that claim requiring an event queue that determines the specific order of each specified device handler.

Claim 52 shall stand or fall by itself, that claim reciting that an API is dynamically linked from the system handler.

c) Claims 13, 14, 18-27 and 49 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Fullerton (U.S. Patent No. 4,607,844).

Patentability shall stand or fall with the patentability of claim 13.

d) Claims 15 and 49 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Fullerton (U.S. Patent No. 4,607,844) (as above) when further considered with Pascal et al. (U.S. Patent No. 5,791,851).

Patentability shall stand or fall with the patentability of claim 15.

e) Claims 28-33 have been rejected under 35 U.S.C. 103(a) as unpatentable over Houriet, Jr. et al. (U.S. Patent No. 5,575,717) in view of Mitchell et al. (U.S. Patent No. 5,872,973).

Patentability shall stand or fall with the patentability of claim 28.

f) Claims 7, 8, and 14 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of David A. Rusling, *The Linux Kernel*, (Hereinafter "Rusling").

Patentability shall stand or fall with the patentability of claim 7.

g) Claims 9, 10, 17, 38, 44 and 47 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Pascal (as applied to claims 22, 23) in further view of Bock (U.S. Patent No. 5,155,856) and Davis (U.S. Patent No. 6,401,208).

Patentability shall stand or fall with the patentability of claim 9.

h) Claims 45, 46 and 50-51 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Wiltshire (U.S. Patent No. 6,409,602).

Patentability shall stand or fall with the patentability of claim 45.

i) Claims 4, 5, 38 and 55-57 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Arbaugh et al. (U.S. Patent No. 6,185,678).

Patentability shall stand or fall with the patentability of claim 4.

j) Claim 6 has been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Arbaugh et al. (U.S. Patent No. 6,185,678) (as above) in further view of Pascal.

Claim 6 shall stand by itself for patentability under this issue.

RESPONSE TO THE REJECTIONS

PRELIMINARY REMARKS CONCERNING THE INVENTION AND THE CLAIMS

It is to be first noted that claims 39-46 have been voluntarily withdrawn by canceling those claims, Applicants reserving the right to prosecute claims to that subject matter on their merits in a later application claiming priority under 35 U.S.C. 120 from this Application.

Additionally, all claims now pending in this application recite the following language or its functional equivalent:

that a “program object calls up an Application Program Interface.”

In contrast, all prior art systems shown in references used in the rejection have passive shared objects that are called through an API. Our claims require an active program object that functions to call a system handler application. In addition to all of the other arguments presented below, which further define issues that establish that the present invention, this limitation, which is present in all remaining claims, defines a structure, method and apparatus that is not disclosed in the prior art used in the rejection of claims in the Office Action mailed March 20, 2003. In addition to the fact that this limitation is not taught by that prior art, this limitation provides definite benefits to the performance of the security for the gaming system and other features of the gaming system as compared to the methods of the prior art references cited in the rejections. The limitation requires that the order of execution in the apparatus, software and games is that the gaming program object calls up the API (Application Program Interface). This is distinct from what is shown in Bunnell (the primary reference in most of the rejections). Bunnell the program objects are passive, are not capable of making calls to the API, and are called by the API by the system handler. This is significantly and unobviously different from the

recited limitation of the API being called by the program objects. This limitation in the claims clearly exists, this limitation is not shown by Bunnell or any other reference used in the rejections of record. There cannot be any motivational basis for changing the order of operation and execution of the functions in Bunnell and the other references used in the rejection. Without any motivational basis for making this significant change, the limitation and the claims containing the limitation cannot be obvious from the art used in the rejection. As that limitation or its substantial equivalent is present in every claim remaining in the Application, all of the rejections of claims are in error and must be withdrawn.

RESPONSE TO REJECTION

b) Claims 1-3, 11-13,16, 19, 29, 31, 34-37, 39-43, 48 and 50-54 are rejected under 35 USC 103(a) as unpatentable over Bunnell (U.S. Patent No. 6,075,939).

The rejection in the Office Action is believed to be fairly summarized as follows:

Bunnell et al. is asserted to show:

- 1) A system handler, executed by the operating system kernel, operable to dynamically link with at least one program object;
- 2) An operating system comprising a system handler;
- 3) A system handler comprising a plurality of device handlers;
- 4) A system handler that loads and executes program devices;
- 5) The kernel is modified to access user level code from ROM;
- 6) The kernel is modified to execute from ROM;
- 7) Kernel modifications are modular;
- 8) The system handler comprises APIs with functions callable from program objects;
- 9) The system handler can manage an event queue; and
- 10) The system handler loads and unloads program objects.

The rejection then lists thirteen (13) elements (identified as b-n elements) in the claims that the Examiner agrees are not shown by Bunnell et al. The rejection then asserts that the elements not disclosed by Bunnell et al. are "typical game element device methods employed on a PC."

In separate rejection under 35 U.S.C. 103(a), the Examiner then cites four references (Rusling, *The Linux Kernel*, <http://www.tldp.org/LDP/tlk/tlk.htm> (1999), Pascal et al. (U.S. Patent 5,971,851), Bock et al. (U.S. Patent No. 5,155,856), and Davis

(U.S. Patent No. 6,401,208) to show individual elements that are recited in dependent claims as obvious over the teachings of Bunnell et al. and the Official Notice taken by the Examiner. These rejections are respectfully traversed.

Claims 1, 2, 16, 29, 31, 35, 48, 50, 51, and 53-54 shall stand or fall with the patentability of claim 1.

The rejection fails in a number of regards. The first level of traversal in the rejection is the failure to appreciate the complexity of the use of a system handler in **A GAMING SYSTEM ENVIRONMENT**, the differences between general PC usage and usage in a gaming system, and the unique aspects of the system recited in the claims as compared to the systems described in the prior art cited against the claims. Additionally, the above amendments to the independent claims emphasize features thought to be implicit in certain claims but now literally and clearly recited in the claims and finding antecedent basis in the original specification as filed. For example, the limitations of:

- a) to verify that the operating system kernel or a code for a shared object has not changed; (e.g., Page 11, lines 14-18)
- b) the gaming program object calling up an Application Program Interface; (Page 6, lines 4-16) and
- c) cause a system handler application load and execute gaming program objects; cause a loaded gaming program object to call up a library of functions; load a first program object from the library, (Page 6, lines 4-16 and Page 10, lines 8-26).

The differences between the gaming system environment and the fields of use described by Bunnell et al. are first evident in the fact that Bunnell et al. does not show a

game controller, game programs, and game objects. There is nothing in Bunnell et al. that directly ties that reference into the field of practice recited in the claims, not only in the preamble, but within functional limitations in the elements of the claims themselves. This substantive initial difference becomes extremely important with regard to the differences in the function and components of the system and methods recited in the claims of the present application.

To begin with, the underlying operations of the claimed system and the system of Bunnell with regard to the terminology of "dynamic linkage" are quite distinct. Claim 1, for example, specifically recites:

"...a system handler application operable to dynamically link with at least one gaming program object..."

This is quite distinct from what is described in Bunnell et al. For example, looking at Figure 4 of Bunnell et al. (and reviewing the entire specification of Bunnell), the operation of the technology does not show a system handler application that "dynamically links with at least one gaming program object..." In fact, the Figure and the disclosure do not show dynamic linking, as recited in the claim, that performs at the same hierarchal level as that recited in the claim. Looking at Figure 4 of Bunnell, the dynamic linking recited in the claim would be above (e.g.—at a higher level, such as at an application level, or between an application and operating system level) all elements shown in that Figure, not at the lower level of the ROM BIOS kernel shown in Figure 4 and described in the text of Bunnell et al. Therefore, in addition to failing to provide any direct nexus into the gaming art, the actual performance of Bunnell et al. is excluded by the present claims, and the present claims require the performance of steps and the presence of recited functions that are not shown by Bunnell.

Reviewing Bunnell in the most positive light, it can be seen that the level of dynamic linkage is in fact substantively different than that recited in the claims, occurring

between the system handler and a gaming program object. For example, looking at the complete disclosure of Bunnell, wherever “dynamic” or “linkage” or “linking” appears in the specific or claims, the following portions of the Bunnell specification should be noted.

At column 7, lines 62-65, Bunnell states under “Alternate components” that there should be a “dynamic system call facility” which does not expand the method of performance specifically taught by Bunnell and as illustrated by Figure 4. This is a general reference, with no probative teaching, and no description of a system handler dynamically linked to a gaming program object.

Column 10, lines 55-67 describes dynamically establishing relationships and then refers to objects that have already been linked. This is a dynamic linking for system calls, dynamically linking system components together. That is different from the type of dynamic linking of the present invention. Figure 4 of Bunnell shows two kernel components linked together, which defines the level of dynamic linkage, in a Posix compliant API. Posix is a standard for operating systems, and Bunnell teaches an implementation of the standard system. Posix has nothing to do with the “dynamic linkage” recited in the claims. Bunnell describes a dynamic system call, not a dynamic linkage. At column 10, lines 60-66, Bunnell merely describes dynamically establishing relationships between global functions and symbols. This is not a description of a system handler dynamically linked to a gaming program object.

Effects of the Limitation on Establishing Non-Obviousness

This is extremely material with respect to the new limitations added to the claims. As noted, the system of Bunnell provides an API directly to applications (see Figures 2 and 4 of Bunnell), and the linking is only between the system handler and the objects. All of the present claims emphasize this fundamental difference by requiring that

the objects call up a function from within the API. This is not a trivial difference. A gaming system that operate in this manner enhances security, which is a primary obligation and requirement of gaming equipment. There is no suggestion in Bunnell for this modification specifically needed in the gaming industry because Bunnell has no concept of the requirements of the gaming industry. Applicants have not noted this specific feature in the four other secondary references cited in later rejections under 35 USC 103(a). As this feature is not disclosed in the references and as this feature is a unique advantage in the gaming environment that is not central to the primary reference (Bunnell), the claims (1-47 and 49-54) reciting this limitation are not obvious from the art cited in the rejection.

The difference of the capability and actual performance at “higher levels” within the software is well understood in the art in general terms, and is spelled out clearly in the practice of the present invention in the dynamic linking of the system handler to the casino gaming objects. Software is written in different hierarchies of content. As one progresses from kernel level, to user API level, to object level etc., the program becomes more abstract moving up the hierarchy of the software. The kernel actually talks directly to hardware, then hardware can use game functions API to execute the gaming performance. This is what is meant when this response refers to a higher level of performance as compared to the software and operating systems of the references, where they do not operate at that level of abstraction.

Claim 48 also recites that, in addition to the novel operation of the software in the execution of data and objects, the system operates to verify that the operating system kernel or a code for a shared object has not changed. Bunnell does not show the combination of the recited execution of software and the verification process. Claims 55-57 also recite this additional step.

In addition, the performance of a dynamic linkage, as recited in these claims, allows for dynamic unlinking. This means that we can unload a game object and replace it with another game object as recited in other claims. This functionality, as recited in claims 5, 19 21, 22, 23, 25, and 26. For example, the recitations in claim 19 includes:

19. A machine-readable medium with instructions thereon, the medium being within a wagering apparatus, the instructions when executed operable to cause a computer to:

cause a system handler application load and execute gaming program objects;

cause a loaded gaming program object to call up a library of functions;

load a first program object from the library,

execute the first program object,

store data variables in nonvolatile storage, such that a second program object in the library later loaded can access the data variables in nonvolatile storage,

unload the first program object, and

load the second program object.

This functionality is not taught, enabled or suggested by Bunnell et al., alone or in combination with all of the secondary references.

Column 11, lines 20-65 of Bunnell describe “Dynamic System Call Management,” which again is not a disclosure of a system handler that dynamically links to a gaming program object. The term “dynamic linkage” as used in the present invention and as described in the specification and understood in the art defines a linkage above the user program level, which is above the kernel level shown in the Figure (4 of

Bunnell et al.) and as described in the specification of Bunnell. In addition, the user program links in the shared objects are gaming specific, which is not shown by Bunnell and is not motivated by the four additional references cited in the rejection. In the practice of the present invention, the dynamic linkage is also used with the API. This is quite distinct from Bunnell's POSIX system call for the API. This is sufficiently distinct as to have warranted the filing of new claims 52-54 to clearly claim that distinct feature.

The five times that dynamic system calls are referred to by Bunnell et al. on column 12 (lines, 4-67) do not in any way contradict the earlier usage or expand on that usage to indicate to one skilled in the art that a system handler can be dynamically linked to a gaming program object. Similarly, the reference on Column 13, lines 22-26 to a Dynamic System Call Table does not in any way improve upon the deficiencies noted with regard to the disclosure of Bunnell. That reference, and none of the secondary references, shows the dynamic linking of a system handler application to a program game object. Nothing equivalent to that has been shown to be taught by Bunnell et al. or any of the secondary references used in the rejection.

It is therefore clear, that even above the thirteen deficiencies that the Examiner has noted in Bunnell, there is another, and even more egregious deficiency that has not been asserted to be obvious in view of the secondary references.

Turning now to the thirteen deficiencies in the Bunnell et al. reference, the mere number of differences that the Office Action attempts to take Official Notice of is in itself an indication of the extent of difference between the teachings of Bunnell et al. and the claimed invention. Additionally, the statement that these features are "typical game device methods implemented on a PC" is challenged, as is the general taking of Official Notice on these limitations. The use of PC's in the gaming industry (as opposed to games) is relatively new. The use of a PC to effect these procedures and provide those functions was not obvious to applicants at the time of filing the Application. The

extensive amount of work needed to enable the practice of these functions on a PC clearly indicates that there is neither obviousness nor ability to take Office Notice of those limitations, particularly within a gaming environment.

It is important to note here the environment of the present invention in the gaming industry. Applicants do not deny that it has been possible to use modern PC's and Operating Systems to write games, as that has been accomplished by the inventors. However, it was not obvious or instructed in the prior art or available in inherent components in available PC's and Operating Systems to meet Gaming Regulations. Additionally, gaming systems require significant and critical security in their communication systems, both internally and with outside sources of communication. Failure to provide such security would fail to meet both Gaming Commission requirements and the security needs of casino operators. No such communication security is generally required or used in arcade games. As described and enabled in the specification, extensive modifications, enhancements and safeguards were needed to effect gaming and system designs that could meet the stringent regulation standards, and adapt PC's to the casino environment. Accomplishing this result with a system handler dynamically linked to gaming program objects was not taught by any of the references cited in the rejection, either alone or in combination. The system handler as recited in the claims also provides the API to gaming program objects for security reasons. This is quite distinct from dynamic linkages that have been used in prior art systems of record in the art used in the rejection (if any) in which a program object provides an API to the Application (as done by Bunnell). The procedure of the invention is not taught by Bunnell.

The prior art of record also does not recognize the benefits of the dynamic linking recited in the practice of the invention. Because objects are loaded and then unloaded before the next object is loaded, resources in the memory are preserved, which is always

an issue in the use of computer-based systems. This can enable the system to work more efficiently and more rapidly because of this dynamic linking and memory space utilization.

Claim 3 shall stand or fall with itself for patentability, that claim specifically reciting loading and unloading and executing specific functions within a gaming environment.

It is to be noted in the examination of this claim that three specific functions are recited to be performed by the system handler software in the gaming apparatus: a) unloading a previously loaded gaming program; b) loading a new gaming program; and c) executing the new gaming program object. This function embedded in the systems handler software is unique to the practice of the present invention. As noted above, the teachings of Bunnell do not show such sequential loading from the operation of the system handler. There has also been no specific showing on the record of this prosecution with respect to the limitations of these claims. Claim 3 has been generally linked into a broad rejection, but no specific teaching of the limitations recited herein have ever been shown in the combination of references, and Applicants have not found them.

As these limitations are unique to the present system and have not been shown to be taught in the prior art, the claim must be unobvious. The burden is on the PTO to specifically show that the limitations in the claims as recited are at least specifically suggested and motivated for inclusion into the system to assert obviousness. That has not been done on the record. In this absence, the rejection is clearly in error and must be withdrawn.

Claims 11-12 shall stand or fall with the patentability of claim 11, which recites calling specific gaming programs from an API.

As noted above, Bunnell alone or in combination with the various references **does not show calling any programs from an API**. This has been specifically and repeatedly established in the arguments regarding the patentability of claim 1. These claims recite that the system handler comprises the API with functions (in the API) callable from the gaming program objects. This is essentially the direct opposite of what is shown by the art in this rejection. There is no basis from the references suggested in the rejection for modifying the fundamental relationship between an API, game objects, and calling functions. These claims recite that fundamental difference, and these claims therefore recite subject matter that is not even considered in the art used in this rejection within the gaming industry. With a total and complete absence in the records of this fundamental basis of operation between the systems handler, API and game objects, the rejection has absolutely no foundation in fact and must be reversed.

Claim 13 shall stand or fall by itself, that claim reciting a specific procedure in the performance of the process by first calling functions from the API, loading a first program, executing the first program, unloading the first program, then loading a second program.

This claim is a method equivalent of the apparatus of claim 3, except the specific performance of the steps of loading, unloading and executing gaming functions is specifically recited as a positive step, and not merely as a capability in the system. The rejection of record (e.g., page 6 of the last Office Action) merely regards this limitation as a system handler loading and executing program objects. The claim further recites that the functions be provided in and that the functions are called by the program objects. There is not even an assertion in the rejection that the functions are called from the API by the program objects, and the analysis of claim 1 patentability above establishes that

such a process step is not taught or suggested or motivated by the teachings of the references. This rejection is clearly in error and must be reversed.

Claim 19 shall stand or fall by itself.

In addition to all of the limitations of claim 13, this claim additionally requires that the second (and therefore different function) also be called from within the API. As there is no teaching for the calling of a single function be called from within the API, there can be even less possibility that the references can be found to load a function from the API, unload the function, load a second function from the API, and then execute the second function which has been called from the API by the program object. This entire direction of technology and operation is absent from the teachings of the references and a rejection cannot be sustained. This rejection must be reversed.

Claims 34 and 36-37 shall stand or fall with the patentability of claim 34, that claim requiring an event queue that determines the specific order of each specified device handler.

It must be recalled that the limitations of claim 34 do not only recite that the system handler comprises an event queue determining the order of execution of each specified device handler, but that this operation be performed within the system of claim 1 wherein the functions are called from the API by the gaming object.

Claim 52 shall stand or fall by itself, that claim reciting that an API is dynamically linked from the system handler.

There is no specific teaching in the art that the API is dynamically linked from the system handler. This rejection, as with so many of the earlier rejections is based upon similarities in words from the references as opposed to teachings of the obviousness of

the claim language in the motivational teachings of the references. The references show an API and game functions, so the rejection overlooks the limitations and requirements in the claims that functions must be callable from the API by gaming objects. There is not even a statement or assertion that such a recited performance structure is obvious, but rather the rejection completely overlooks the details of the claim recitation and by inference appears to assume that all relationships are obvious. This is not a supportable basis for a rejection under 35 USC 103(a). There must be some teaching in the art that specific properties, components and functions are to be arranged in a gaming device in the manner that they are recited in the claims on appeal. That has not been done in this rejection. The rejection is clearly in error and must be reversed.

c) Claims 13, 14, 18-27 and 49 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Fullerton (U.S. Patent No. 4,607,844).

Claims 13, 14, 18-27 and 49 shall stand or fall with the patentability of claim 13.

Even if the asserted basis for the citation of Fullerton as set forth in the Office Action is completely accurate (e.g., Fullerton is cited as showing “a gaming machine that stores data variables in non-volatile storage to increase security due to tampering or loss of power...”), Fullerton does not overcome the fundamental deficiency that Bunnell and Fullerton do not show or provide any motivational basis for having the API called up from the gaming objects. In the absence of that showing, there can be no obviousness of these claims.

d) Claims 15 and 49 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Fullerton (U.S. Patent No. 4,607,844) (as above) when further considered with Pascal et al. (U.S. Patent No. 5,791,851).

Claims 15 and 49 shall stand or fall with the patentability of claim 15.

Insofar as Pascal is cited to show that alternative operating systems that have call back functions in a **game machine** (not gaming machine), that teaching is accepted. However, Pascal provides no teaching whatsoever to overcome each and every one of the deficiencies noted above with respect to the teachings of Bunnell alone or Bunnell in view of Fullerton. The rejection of claims 15 and 49 must therefore fail for at least the reasons that the lack of teaching of the recited and claimed elements of the claims from which these claims are dependent are not shown by Bunnell and those deficiencies are not overcome by Pascal.

e) Claims 28-33 have been rejected under 35 U.S.C. 103(a) as unpatentable over Houriet, Jr. et al. (U.S. Patent No. 5,575,717) in view of Mitchell et al. (U.S. Patent No. 5,872,973).

Claims 28-33 shall stand or fall with the patentability of claim 28.

These references fail to show the specific features that have been added to these claims regarding calling up an API from the gaming objects. All of the arguments presented above with respect to the Bunnell reference (and Bunnell combined with one or more other references) are applicable to this rejection in the same way. The fact that no reference of record shows the function of this limitation absolutely establishes that the claimed invention is unobvious over the art used in the rejections against the claims.

f) Claims 7, 8, and 14 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of David A. Rusling, *The Linux Kernel*, (Hereinafter "Rusling").

Claims 7, 8 and 14 shall stand or fall with the patentability of claim 7.

Insofar as Rusling is cited to show that Linux is a useful and advantageous operating system, that teaching is accepted. However, Rusling provides no teaching

whatsoever to overcome each and every one of the deficiencies noted above with respect to the teachings of Bunnell alone. The rejection of claims 7, 8 and 14 must therefore fail for at least the reasons that the lack of teaching of the recited and claimed elements of the claims from which these claims are dependent are not shown by Bunnell and those deficiencies are not overcome by Rusling.

g) Claims 9, 10, 17, 38, 44 and 47 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Pascal (as applied to claims 22, 23) in further view of Bock (U.S. Patent No. 5,155,856) and Davis (U.S. Patent No. 6,401,208).

Claims 9, 10, 17, 38, 44 and 47 shall stand or fall with the patentability of claim 9.

As noted above with respect to Pascal, the fundamental deficiencies and lack of teaching of limitations in the independent claims found in Bunnell are not corrected by Pascal. The citation of Bock and Davis for their specific elements of disclosure for dependent claims do not correct the deficiencies that were not addressed by Bunnell in view of Pascal. These claims are therefore patentable because of their ultimate dependence from unobvious independent claims. Even assuming that the teachings of Bock and Davis are accurate for what is asserted, they do not overcome the earlier deficiencies and cannot establish obviousness of the dependent claims.

Additionally, Bock et al. has been cited as showing zeroing-out unused registers, asserting that is what is recited in claims 9, 17, 44, and 47. That is not equivalent to what is done in the present invention by zeroing out memory. The teaching of Bock et al. asserted by the Office Action to describe a method of zeroing out unused registers to provide security during booting up is not equivalent to "...an operating system kernel that is customized to disable selected device handlers..." recited in the claims and the disabling of selected device handlers. Those are immaterial to the booting up security

described by Bock et al. Even more importantly, the methods described in Bock et al. are implemented by hardware. The device handlers recited in the claims of the present invention which zero out memory are implemented in software. This means that in the practice of the invention recited in the claims, there is a CPU that fetches instructions from memory, and it is those instructions that causes the CPU to zero out memory. This zeroing out is not performed during boot-up, but during actual operation of the machine.

h) Claims 45, 46 and 50-51 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Wiltshire (U.S. Patent No. 6,409,602).

Claims 45-46 and 50-51 shall stand or fall with the patentability of claim 45.

Similar to the arguments that the secondary references (Pascal, Rusling, Davis, and Bock) do not either overcome the deficiencies of Bunnell, Wiltshire does not teach the fundamental limitations of these claims and therefore cannot improve or correct the earlier rejection.

i) Claims 4, 5, 38 and 55-57 have been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Arbaugh et al. (U.S. Patent No. 6,185,678).

Claims 4, 5, 38 and 55-57 shall stand or fall with the patentability of claim 4.

Again, even if Arbaugh does provide a sufficient teaching for the elements of the claims for which it has been cited, Arbaugh does not correct the deficiencies of Bunnell which have been extensively discussed above. As the underlying failures of Bunnell have not been overcome by the teachings of Arbaugh, the rejection still remains in error for the reasons cited above.

j) Claim 6 has been rejected under 35 USC 103(a) as unpatentable over Bunnell in view of Arbaugh et al. (U.S. Patent No. 6,185,678) (as above) in further view of Pascal.

Claim 6 shall stand or fall by itself.

Insofar as Pascal is cited to show that alternative operating systems that have call back functions in a **game machine** (not gaming machine), that teaching is accepted. However, Pascal provides no teaching whatsoever to overcome each and every one of the deficiencies noted above with respect to the teachings of Bunnell alone or Bunnell in view of Arbaugh. The rejection of claim 6 must therefore fail for at least the reasons that the lack of teaching of the recited and claimed elements of the claims from which these claims are dependent are not shown by Bunnell and Arbaugh and those deficiencies are not overcome by Pascal.

CONCLUSION AND SUMMARY OF ARGUMENTS

The primary reference not only fails to establish utility of the teachings of that reference in the field of gaming, but also, in addition to the thirteen elements and limitations of the invention acknowledged by the Office Action to be missing from the Bunnell et al. reference, that reference fails to describe, suggest or enable a system handler to dynamically link to at least one gaming program object. The four secondary references fail to teach these limitations and fail to teach the specific limitations for which they were cited in the Office Action.

CONCLUSION

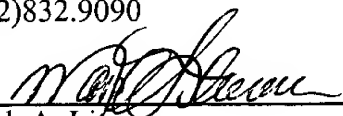
All rejections of record have been shown in detail to be in error. The rejection should be reversed and all claims should be indicated as allowable.

Applicants believe the claims are in condition for allowance and request reconsideration of the application and allowance of the claims. The Examiner is invited to telephone the below-signed attorney at 952-832-9090 to discuss any questions that may remain with respect to the present application.

Respectfully submitted,
INVENTOR NAMES

By their Representatives,
MARK A. LITMAN & ASSOCIATES, P.A.
York Business Center, Suite 205
3209 West 76th Street
Edina, MN 55435
(952)832.9090

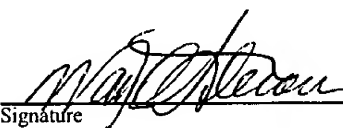
Date: FEBRUARY 9, 2004 By



Mark A. Litman
Reg. No. 26,390

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope with first class postage prepaid and addressed to MAIL STOP: APPEALS, P.O. Box 1450, Commissioner for Patents, Alexandria, VA 22313-1450 on February 9, 2004.

Name: Mark A. Litman



Signature

APPENDIX - THE CLAIMS ON APPEAL

1. (PREVIOUSLY AMENDED) A computerized wagering game apparatus, comprising:

a computerized game controller having a processor, memory, and nonvolatile

storage, the computerized game controller being operable to control a computerized wagering game;

an operating system comprising: a system handler application operable to

dynamically link with at least one gaming program object;

an Application Program Interface having functions callable from the gaming program object and

an operating system kernel that executes the system handler application.

2. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the system handler application comprises an event handler.

3. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the system handler application comprises software having the ability when executed to:

unload a previous gaming program object if a previous object has been loaded;

load a new gaming program object; and

execute the new gaming program object.

4. (PREVIOUSLY AMENDED) The computerized wagering game apparatus of claim 1, wherein data variables modified by the gaming program objects are stored in nonvolatile storage, and functions verify that the operating system or code for a shared object has not changed.

5. (ORIGINAL) The computerized wagering game apparatus of claim 4, further comprising a game state device providing a variable name index to associated variable data locations within the nonvolatile storage.

6. (PREVIOUSLY AMENDED) The computerized wagering game apparatus of claim 4, wherein changing a data variable in nonvolatile storage causes execution of a corresponding callback function in the system handler application gaming program object.

7. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the computerized game controller comprises an IBM PC-compatible computer.

8. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the operating system kernel is a Linux operating system kernel.

9. (ORIGINAL) The computerized wagering game apparatus of claim 8, wherein the Linux operating system kernel has at least one selected device handler disabled.

10. (ORIGINAL) The computerized wagering game apparatus of claim 9, wherein the at least one selected device handler that is disabled is selected from the group consisting of a keyboard handler, an I/O port handler, a network interface handler, a storage device controller handler, and a I/O device handler.

11. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the system handler application comprises an API with functions callable from the gaming program objects.

12. (ORIGINAL) The computerized wagering game apparatus of claim 11, wherein the API comprises functions that are specific to a computerized gaming apparatus.

13. (PREVIOUSLY AMENDED) A method of managing data in a computerized wagering game apparatus via a system handler application, comprising:

loading a first program object and providing an Application Program Interface having functions called by the first program object,

executing the first program object,

storing data variables in nonvolatile storage, such that a second program object later

loaded can access the data variables in nonvolatile storage,

unloading the first program object, and

loading a second program object.

14. (ORIGINAL) The method of claim 13, further comprising executing a corresponding callback function upon alteration of variable data in nonvolatile storage.

15. (ORIGINAL) The method of claim 13, further comprising handling events via the system handler application.

16. (PREVIOUSLY AMENDED) A secure computerized wagering game system controlled by a general-purpose computer, comprising an operating

system kernel that is customized to disable selected device handlers, and a program that loads a first program game object and the first program game object calls up a function from within an Application Program Interface.

17. (PREVIOUSLY AMENDED) A secure computerized wagering game system controlled by a general-purpose computer comprising nonvolatile storage that stores program variables, such that loss of power does not result in loss of the state of the computerized wagering game system, and a program that loads a first program game -object and the first program game object calls up a function from within an Application Program Interface.

18. (ORIGINAL) The secure computerized wagering game system of claim 17, further comprising at least one gaming program object, such that a single gaming program object is loaded and executed at any one time but gaming program objects are operable to share data via the program variables in nonvolatile storage.

19. (PREVIOUSLY AMENDED) A machine-readable medium with instructions thereon, the medium being within a wagering apparatus, the instructions when executed operable to cause a computer to:

cause a system handler application to load and execute gaming program objects;

cause a loaded gaming program object to call up a library of functions provided by the system handler application;

load a first program object from the library,

execute the first program object, including having the first program object call up a function from within an Application Program Interface,

store data variables in nonvolatile storage, such that a second program object in the library later loaded can access the data variables in nonvolatile storage,

unload the first program object, and

load the second program object, including having the second program object call up a function from within an Application Program Interface.

20. (ORIGINAL) The machine-readable medium of claim 19, with further instructions operable when executed to cause a computer to execute a corresponding callback function upon alteration of variable data in the nonvolatile storage.

21. (ORIGINAL) The machine-readable medium of claim 19, with further instructions operable when executed to cause a computer to manage events via the system handler application.

22. (PREVIOUSLY AMENDED) A machine-readable medium with instructions thereon, the instructions when executed operable to cause a computer to manage at least one gaming program object via a system handler application, the gaming program object calling up a function from within an Application Program Interface such that a single gaming program object is loaded and executed at any one time but gaming program objects are operable to share data via the program variables in a nonvolatile storage.

23. (ORIGINAL) The machine-readable medium of claim 22, with further instructions operable when executed to cause a computer to provide functions through an API that comprises a part of the system handler application.

24. (PREVIOUSLY AMENDED) A machine-readable medium with instructions thereon, the instructions when executed operable to cause a computer to manage at least one gaming program object via a system handler application, such that a single gaming program object is executed at any one time, wherein gaming program objects are operable to share data in nonvolatile storage and to call up a function from within an Application Program Interface.

25. (ORIGINAL) The machine-readable medium of claim 21, wherein only one gaming program object executes at any one time.

26. (PREVIOUSLY AMENDED) The machine-readable medium of claim 21, with further instructions operable when executed to cause a computer to provide functions through the API that comprises part of the system handler application.

27. (PREVIOUSLY AMENDED) A machine-readable medium with instructions thereon, the instructions when executed are operable to store game data in non-volatile storage, such that the state of the computerized wagering game system is maintained when the machine loses power, and wherein gaming program objects are operable to share data in the nonvolatile storage and to call up a function from within an Application Program Interface.

28. (PREVIOUSLY AMENDED) A gaming machine architecture, comprising an operating system, and a plurality of shared objects; wherein each shared object describes game personality in a selected mode, and wherein gaming program objects are operable to share data in nonvolatile storage and to call up a function from within an Application Program Interface.

29. (ORIGINAL) The gaming machine architecture of claim 28, wherein the operating system comprises an IBM PC-based operating system.

30. (ORIGINAL) The gaming machine architecture of claim 28, wherein the operating system comprises a system handler.

31. (ORIGINAL) The gaming machine architecture of claim 30, wherein the system handler comprises a plurality of device handlers.
32. (ORIGINAL) The gaming machine architecture of claim 30, wherein the system handler comprises an event queue.
33. (PREVIOUSLY AMENDED) The gaming machine architecture of claim 30, wherein the system handler comprises a plurality of API callable functions, callable by the shared objects.
34. (ORIGINAL) The apparatus of claim 1, wherein the system handler comprises an event queue that determines the order of execution of each specified device handler.
35. (ORIGINAL) The apparatus of claim 1, wherein the system handler comprises an API having a library of functions.
36. (ORIGINAL) The apparatus of claim 34, wherein the event queue is capable of queuing on a first come, first serve basis.
37. (ORIGINAL) The apparatus of claim 34, wherein the event queue is capable of queuing using more than one criteria.
38. (ORIGINAL) The apparatus of claim 1, wherein the system handler and kernel work in communication to hash the system handler code and the operating system kernel code.

39. (CANCELLED)

40. (CANCELLED)

41. (CANCELLED)

42. (CANCELLED)

43. (CANCELLED)

44. (CANCELLED)

45. (CANCELLED)

46. (CANCELLED)

47. (PREVIOUSLY AMENDED) A method of modifying an operating system kernel in a gaming apparatus where a program object calls up a function from within an Application Program Interface, comprising at least one modification to obtain functionality selected from the group consisting of:

- 1) accessing user level code from ROM;
- 2) executing user level code from ROM;
- 3) zeroing out unused RAM;

4) testing and/or hashing the kernel to verify that the operating system kernel or a code for a shared object has not changed;

5) disabling selected device handlers.

48. (ORIGINAL) The computerized wagering game of claim 1 wherein the operating system kernel executing on the computerized game controller comprises an element of a universal operating system also comprising a system handler.

49. (ORIGINAL) The computerized wagering game of claim 1 wherein the non-volatile storage is controlled by a general-purpose computer, the non-volatile storage stores game data, and the storage of game data in the non-volatile storage preserves the state of the computerized wagering game system upon loss of power.

50. (ORIGINAL) The computerized wagering system of claim 1 operating within a networked on-line system.

51. (ORIGINAL) The computerized wagering system of claim 1 wherein the system controls a progressive meter.

52. (PREVIOUSLY AMENDED) The computerized wagering system of claim 1 wherein an Application Program Interface is also dynamically linked from the systems handler.

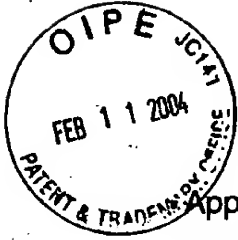
53. (PREVIOUSLY AMENDED) The machine readable medium of claim 19 wherein the instructions when executed are operable to dynamically link an Application Program Interface to a gaming program object.

54. (PREVIOUSLY AMENDED) The machine readable medium of claim 22 wherein the instructions when executed are operable to dynamically link an Application Program Interface to a gaming program object.

55. (PREVIOUSLY ADDED) The computerized wagering game apparatus of claim 1 wherein the operating system operates a function to verify that the operating system kernel or a code for a shared object has not changed.

56. (PREVIOUSLY ADDED) A method of managing data in a computerized wagering game apparatus via a system handler application according to claim 1 wherein a function is performed to verify that an operating system kernel or a code for a shared object has not changed.

57. (PREVIOUSLY AMENDED) The computerized wagering game apparatus of claim 1 with instructions thereon wherein a function is performed to verify that an operating system kernel or a code for a shared object has not changed



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

AF
3713

Applicant: Michael G. Martinek et al. Examiner: S. Ashburn
Serial No. 09/520,405 Group Art Unit: 3713
Filed: March 8, 2000 Docket No. PA0390.ap.US
Title: COMPUTERIZED GAMING SYSTEM METHOD AND APPARATUS

MAIL STOP APPEAL BRIEF - PATENTS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RECEIVED

FEB 17 2004

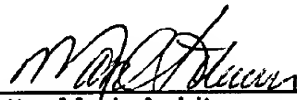
TECHNOLOGY CENTER R3700

The following documents are hereby submitted:

- ☒ Appeal Brief to the Board of Patent Appeals and Interferences of the United States Patent and Trademark Office (three copies)
- ☒ Authorization to withdraw \$165.00 to cover Appeal Brief Fee of a small entity
- ☒ Transmittal Sheet
- ☒ Return postcard

Please consider this a PETITION FOR EXTENSION OF TIME for sufficient number of months to enter these papers if an additional extension of time is deemed necessary by the Office. Authorization is hereby given to charge Deposit Account Number 50-1391 if such additional extension is necessary.

MARK A. LITMAN & ASSOCIATES, P.A.
York Business Center, Suite 205, 3209 W. 76th St.
Edina, MN 55435 (952-832-9090)

By: 
Atty: Mark A. Litman
Reg. No. 26,390

CERTIFICATE UNDER 37 C.F.R. 1.8: The undersigned hereby certifies that this Transmittal Letter and the paper, as described herein, are being deposited in the United States Postal Service, as first class mail, with sufficient postage, in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450 on February 9, 2004.

Mark A. Litman
Name


Signature